## REMARKS

Favorable consideration of this application is respectfully requested.

Claims 1-42 are currently active in this case. Claims 29-42 have been added by way of the present amendment. Each new claim is supported by the specification and claims as originally submitted and no new matter has been added.

Applicants respectfully traverse the rejection of Claim 1 under 35 USC 102(b) as being anticipated by *Lavagno* and under 35 USC 102(a) as being anticipated by *Polis*. Claim 1 recites:

> *1. (Original) In a system-level design and integration system which facilitates composing behavior, capturing architecture and mapping behavior onto architecture, a method for specification of scheduling for simulation and implementation of consumer embedded systems; the method comprising the steps of:*
>
> *a) identifying the schedulers in an architecture;*
>
> *b) identifying the schedulables constituting behavior;*
>
> *c) assigning schedulables and schedulers to each other;*
>
> *d) arranging for the schedulers to find their respective assigned schedulables;*
>
> *e) arranging for the schedulables to find their respective assigned scheduler;*
>
> *f) sending an event to a schedulable;*
>
> *g) sending an activation notice from the schedulable in step f) to its assigned scheduler;*
>
> *h) sending a message from the assigned scheduler of step g) to the schedulable of step f) to start its behavior's reaction; and*
>
> *I) sending a finish notice from the schedulable of step f) to the scheduler of step g) when said reaction is completed.*

However, *Lavagno* and *Polis* fail to teach or suggest similar subject matter.

As a preliminary matter, Applicants respectfully note that the *Lavagno* publication provided with the outstanding Office action is only a series of slide presentations with bullet points, but has no cohesive written description tying the those slides together in a manner that teaches the claimed invention. Most notably, *Lavagno* does not teach the steps of identifying schedulers in an architecture and assigning schedulables to schedulers.

Applicants respectfully traverse any assertion that would equate *Lavagno's* "System Architecture" (*Lavagno*, page 8), particularly the e2, e3, and e4 signals from various Hardware partitions to and/or from various ports. Applicants respectfully submit that the proposed signals of *Lavagno* do not teach or suggest identification of schedulables or assigning schedulables to schedulers. At most, *Lavagno* page 8 shows a communication from hardware to CFSMs, but provides no indication that that communication is either an identification or an assignment of schedulables.

Furthermore, Applicants respectfully note that *Lavagno's* e2, e3, and e4 signals are input and/or outputs to CFSM's and do not appear to interact with the scheduler (*Lavagno's* scheduler, page 8, is not connected to the CSFM's). It may be that the scheduler controls execution of the CFSMs, but that is not shown, and, more importantly, such an interaction is not the same as identifying and assigning schedulables. In fact *Lavagno*, page 8, appears to show a fixed allocation of the signals (whether or not they are schedulables) between CFSMs.

Nevertheless, without the underlying presentation associated with the *Lavagno* slides, Applicants are unable to fully address any rejection based thereon. Therefore, Applicants respectfully submit that *Lavagno* be withdrawn as a reference.

If the reference is maintained, since *Lavagno* does not teach or suggest specific limitations in Applicants claimed invention, Applicants respectfully submit that Claim 1 cannot be anticipated by *Lavagno*. Accordingly, Applicants respectfully submit that Claim 1 is patentable over *Lavagno*.

Applicants respectfully note that *Polis* describes a Hardware Software Co-Design system, that uses high level design specifications (e.g., page 31, 3.1, 1. High Level Language Translation), including classical Finite State Machines (CFSMs), and System Behavior information (page 32, Fig. 7) to allegedly perform partitioning between software and hardware components of a design (page 31, section 3.1, 4. Design Partitioning). The partitioned design is then allegedly synthesized for both hardware and software partitions of the design (Fig. 7, page 32, SW Synthesis & HW Synthesis).

However, Applicants respectfully note that *Polis* does not teach partitioning as shown in Fig. 7. In fact, *Polis* does not provide any Hardware/Software partitioning (*Polis*, page 85, line 1). That *Polis* might provide a flexible environment that supports hand-partitioning or embedded methods for automatic partitioning does not teach applicants claimed invention. That *Polis* includes hardware partitioning (page 85, footnote 36), is not sufficient to support a system with partitioning between SW and HW components.

Applicants respectfully traverse the assertion in the outstanding Office Action that states *Polis'* Scheduler Template + Timing Constraints (Fig. 7, page 32) teaches Applicants step of *"identifying schedulers in an architecture."* Applicants admit that *Polis* discusses a scheduling template that is applied to an OS synthesis step in the *Polis* system. However, Applicants claimed step of *"identifying schedulers in an architecture,"* is entirely different from the use of a scheduling template. At best, a template requires schedulers that fit the template. More importantly, a scheduling template alone does not teach or suggest identification of schedulers in an architecture.

Applicants also respectfully traverse the assertion that *Polis* teaches Applicants claimed step of "identifying schedulables constituting behavior." Applicants respectfully traverse any assertion that might equate *Polis'* Fig. 7 to a step of identifying schedulables constituting behavior. As a preliminary matter, as shown in Fig. 7, *Polis'* System Behavior is used for Partitioning and Verifying Intermediate Format. However, as noted above, *Polis* does not provide partitioning in the system, and, neither partitioning nor verification of intermediate formats are an identification of schedulers.

Further, Applicants respectfully traverse the assertion that *Polis* teaches Applicants claimed step of "assigning schedulables and schedulers to each other." Applicants respectfully assert that no part of *Polis'* Fig. 7 suggests an assignment of schedulables to schedulers. Further, *Polis'* only examples of scheduling are only described as being performed by a single scheduler. However, Applicants' claimed invention uses schedulers (plural).

Applicants respectfully traverse the assertion in the outstanding Office Action that states that *Polis* teach "g) sending an activation notice from the schedulable in step f) to its assigned scheduler of step g)." However, *Polis* only describes a single scheduler. More importantly, *Polis* does not send activation notices between schedulables and schedulers. In fact, *Polis* only describes a task based loop (e.g., priority based; see page 24, 2.2. dynamic or run time), or round robin scheduler that does not utilize activation notices.

Since *Polis* does not describe subject matter specifically claimed in Claim 1 (e.g., identifying schedulers in an architecture; assigning schedulables to schedulers; or activation notices between schedulables and schedulers), Applicants respectfully submit that Claim 1 cannot be anticipated by *Polis*. Accordingly, Applicants respectfully submit that Claim 1 is patentable over the cited references.

Applicants respectfully traverse the rejection of Claim 15 under under 35 USC 102(b) as being anticipated by *Lavagno* and under 35 USC 102(a) as being anticipated by *Polis*. Claim 2 recites:

> ***2. (Original) The method recited in claim 1 further comprising the steps of:***
>
> ***h1) having the scheduler of step g) determine whether the reaction of the schedulable of step f) must be preempted before it is finished;***
>
> ***h2) if in step h1) preemption is required, sending a suspend message from the scheduler of step g) to the schedulable of step f) to temporarily halt the execution of its reaction;***
>
> ***h3) if in step h1) preemption is required, causing the schedulable of step f) to halt its reaction in response to the suspend message;***
>
> ***h4) if in step H1) preemption is required, eventually sending a resume message from the scheduler of step g) to the schedulable of step f).***

However, *Lavagno* and *Polis* fail to teach or suggest similar subject matter.

Applicants again respectfully request withdrawal of the *Lavagno* reference because the slides do not fully teach or suggest the claimed method, and, absent the underlying presentation, Applicants are unable to fully address any rejections based thereon.

Regarding *Polis*, Applicants respectfully traverse the assertion in the outstanding Office Action that states *Polis'* Scheduler Template + Timing Constraints (Fig. 7, page 32) teaches Claim 15's step of *"identifying schedulers in an architecture."* As noted above, the claimed step of *"identifying schedulers in an architecture,"* is entirely different from the use of a scheduling template.

Applicants also respectfully traverse the assertion that *Polis* teaches Applicants claimed step of "identifying schedulables constituting behavior." Again, *Polis'* System Behavior is used for Partitioning and Verifying Intermediate Format. However, as noted above, *Polis* does not provide partitioning in the system, and, neither partitioning nor verification of intermediate formats are an identification of schedulers.

Further, *Polis'* only examples of scheduling are only described as being performed by a single scheduler, and no part of *Polis'* Fig. 7 suggests an assignment of schedulables to schedulers. And, *Polis* does not send activation notices between schedulables and schedulers. In fact, *Polis* only describes a task based loop (e.g., priority based; see page 24, 2.2. dynamic or run time), or round robin scheduler that does not utilize activation notices.

Since *Polis* does not describe subject matter specifically claimed in Claim 15 (e.g., identifying schedulers in an architecture; assigning schedulables to schedulers; or activation notices between schedulables and schedulers), Applicants respectfully submit that Claim 15 cannot be anticipated by *Polis*. Accordingly, Applicants respectfully submit that Claim 15 is patentable over the cited references.

Regarding the rejections of Claims 2-6 and 16-20, Applicants respectfully traverse the rewriting of the claims. Applicants respectfully note that it is unclear why the claims were re-written and what the re-written claims are, so it is impossible for Applicants to properly address the rejection. It appears that the claims have been re-written in a broader form. Applicants respectfully submit that, absent amendment by Applicants, limitations present in the original claims cannot be written out of the claims. Nevertheless, Applicants respectfully note that a preemption of a currently executing schedulable must be a preemption of a schedulable that was started by *"sending an activation note from the schedulable to its assigned scheduler."* Since *Polis* fails to teach or suggest the same, Applicants respectfully note that Claim 2 and all claims dependent therefrom are also patentable over the *Polis*.

Applicants respectfully submit new independent Claims 29 and 41. Claim 29 recites:

> ***29. (New) In a system-level design and integration system which facilitates composing behavior, capturing architecture and mapping behavior onto architecture, a method for specification of scheduling for simulation and implementation of consumer embedded systems; the method comprising the steps of:***
>
> ***a) identifying the schedulers in an architecture;***
>
> ***b) identifying the schedulables constituting behavior;***
>
> ***c) assigning schedulables and schedulers to each other;***
>
> ***d) arranging for the schedulers to find their respective assigned schedulables;***
>
> ***e) arranging for the schedulables to find their respective assigned scheduler;***
>
> ***f) sending an event to a schedulable;***
>
> ***g) sending an activation notice from the schedulable in step f) to its assigned scheduler;***
>
> ***h) sending a message from the assigned scheduler of step g) to the schedulable of step f) to start its behavior's reaction;***
>
> ***h1) having the scheduler of step g) determine whether the reaction of the schedulable of step f) must be preempted before it is finished;***

*h2) if in step h1) preemption is required, sending a suspend message from the scheduler of step g) to the schedulable of step f) to temporarily halt the execution of its reaction;*

*h3) if in step h1) preemption is required, causing the schedulable of step f) to halt its reaction in response to the suspend message;*

*h4) if in step h1) preemption is required, eventually sending a resume message from the scheduler of step g) to the schedulable of step f); and*

*i) sending a finish notice from the schedulable of step f) to the scheduler of step g) when said reaction is completed.*

Claim 41 recites:

*41. (New) In a system-level design and integration system which facilitates composing behavior, capturing architecture and mapping behavior onto architecture, a method for specification of scheduling for simulation and implementation of consumer embedded systems; the method comprising the steps of:*

*a) identifying the schedulers in an architecture;*

*b) identifying the schedulables constituting behavior;*

*c) assigning schedulables and schedulers to each other;*

*d) arranging for the schedulers to find their respective assigned schedulables;*

*e) arranging for the schedulables to find their respective assigned scheduler;*

*f) sending an event to a schedulable;*

*g) sending an activation notice from the schedulable in step f) to its assigned scheduler;*

*h) sending a message from the assigned scheduler of step g) to the schedulable of step f) to start its behavior's reaction;*

*h1) having the scheduler of step g) determine whether the reaction of the schedulable of step f) must be preempted before it is finished;*

*h2) if in step h1) preemption is required, sending a suspend message from the scheduler of step g) to the* *schedulable of step f) to temporarily halt the execution of its reaction;*

*h3) if in step h1) preemption is required, causing the schedulable of step f) to halt its reaction in response to the suspend message;*

*h4) if in step H1) preemption is required, eventually sending a resume message from the scheduler of step g) to the schedulable of step f); and*

*i) sending a finish notice from the schedulable of step f) to the scheduler of step g) when said reaction is completed;*

*wherein:*

*each message comprises one of a series of schedulable-scheduler messages comprising a set of messages comprising: (1) an activation message from the schedulable to a scheduler indicating the schedulable wants to react to an event; (2) a start message from the scheduler instructing the schedulable to start; (3) a suspend message sent form the scheduler to suspend the schedulable's reaction; (4) a resume message from the scheduler to the schedulable; and (5) a completion message from the schedulable to the scheduler;*

*the messages are modeled based on a stored value of how long it takes to transition through states specified by the message;*

*the schedulables react to events, and the events comprise a set of events having a timestamp that indicates an order in which a simulator processes the events;*

*the schedulables are activated based on a combination of non fixed frequency activation of the schedulable and a priority assigned to the schedulable;*

*resources utilized to process the schedulables are architectural resources allocated to the schedulables via a contention process;*

*at least one schedulable is a parent scheduler comprising a hierarchy of lesser schedulables including a scheduler configured to implement a scheduler interface; and*

*activation of the parent schedulable comprises an activation that propagates up the hierarchy of schedulables.*

However, the cited references fail to teach or suggest similar subject matter. Accordingly, Applicants respectfully submit that new Claims 29 and 41 are patentable.

New Dependent Claims 30-40 and 42 are also submitted herewith. Each dependent claims further distinguishes each of the cited references.

Based on the patentability of Independent Claims 1, 15, 29, and 41, Applicants further respectfully submit that dependent Claims 2-14, 16-28, 30-40, and 42 are also patentable.

Consequently, no further issues are believed to be outstanding, and it is respectfully submitted that this case is in condition for allowance. An early and favorable action is respectfully requested.

Respectfully submitted,
REED SMITH LLP

Dated: September 7, 2004

By:
Name: John W. Carpenter
Registration No. 39,129

Two Embarcadero Center
Suite 2000
PO Box 7936
San Francisco, CA 94120-7936
***Direct Dial (415) 659-5927***
(415) 391-8269 Facsimile